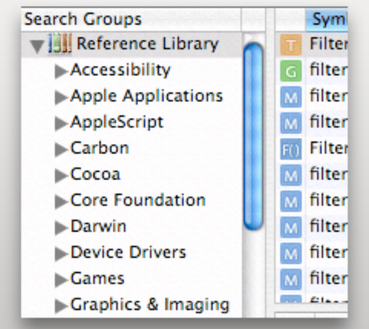
Moving to GCC 4.0

Session 404

These are confidential sessions—please refrain from filming or taking pictures.



Development Tools

Hello, We Have A New Compiler!

Geoffrey Keating
GCC Engineer
geoffk@apple.com

http://developer.apple.com/wwdc2005/

What You'll Learn

- Why should you move to GCC 4.0?
- New features in GCC 4.0
  - You might want to use
- Bugs fixed in GCC 4.0
  - Might require you to fix your code

Intel Architecture Support

Xcode 2.1

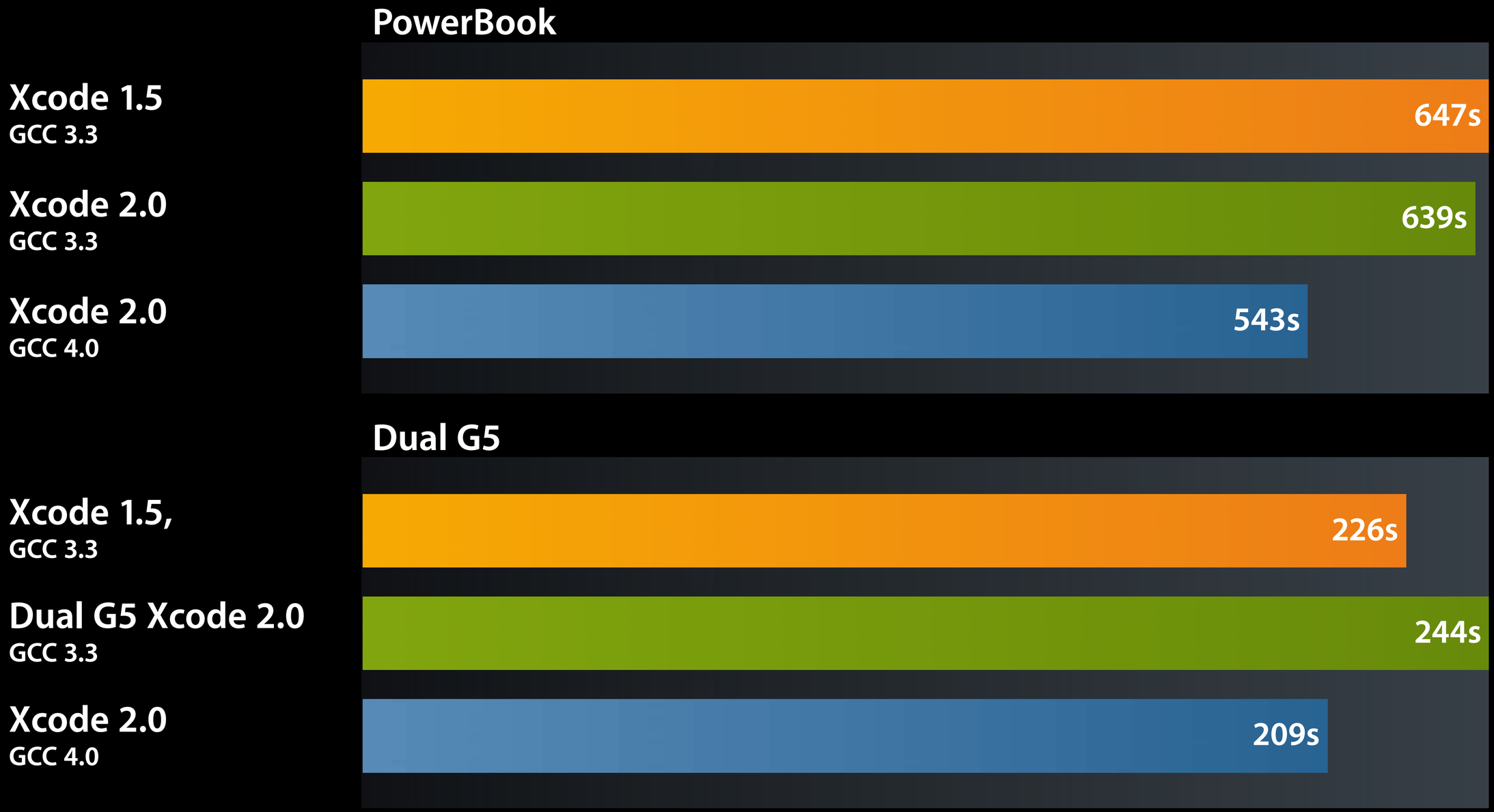
- GCC 4.0 is required for building Intel architecture executables

64-bit Support

- GCC 4.0 is required for building 64-bit executables

Setting	Value
Architectures	ppc ppc64

Quicker Compiler



Better Code Generation

Xcode 2.1

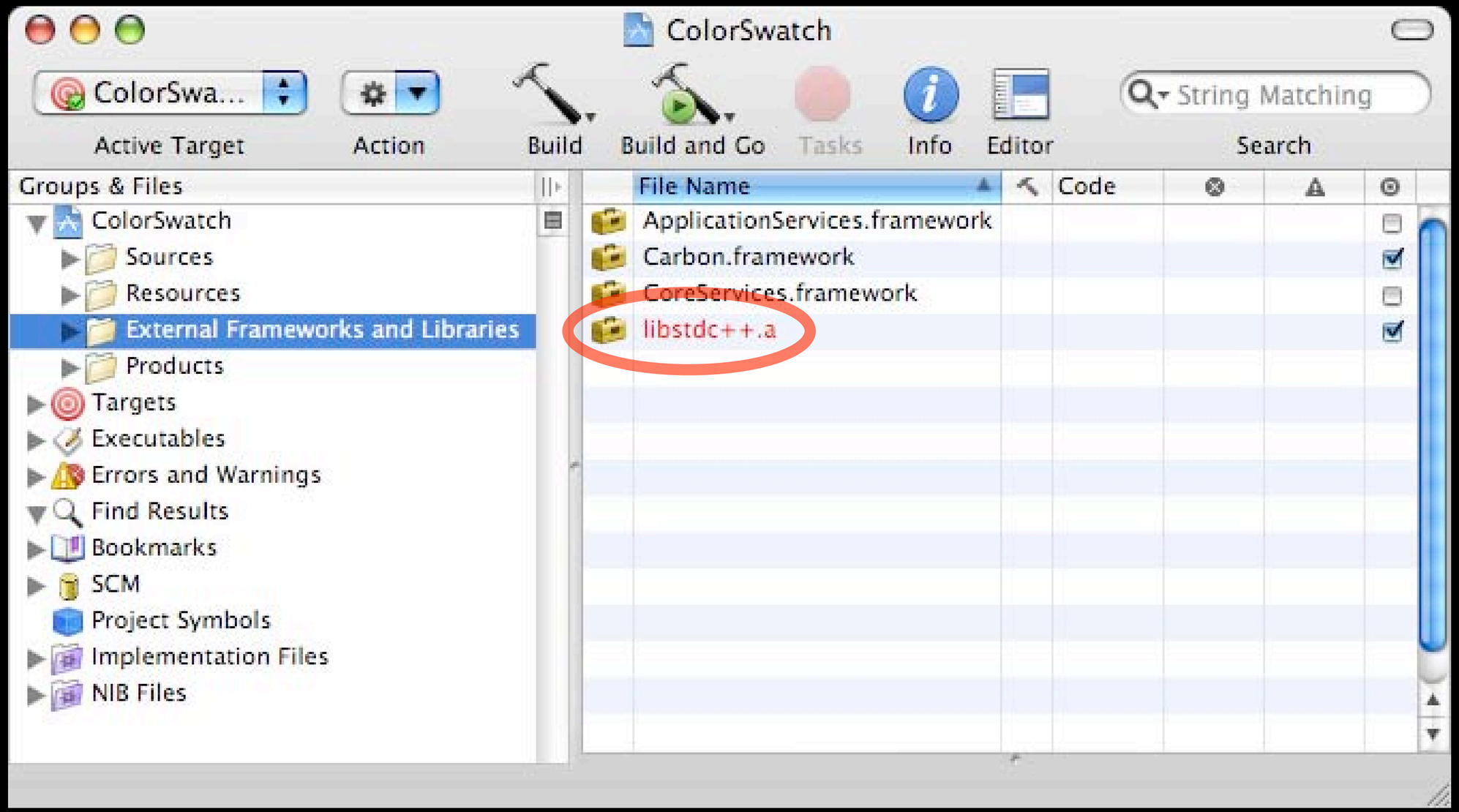
Shared C++ Runtime

- Available in Panther 10.3.9 and later
- Smaller application size
- Faster application loading, switching
- Correctness with streams

Possible Targets on Tiger

	GCC 2.95.1	GCC 3.1	GCC 3.3	GCC 4.0
10.0-10.2.8	Requires SDK	Requires SDK	C++ requires SDK	C, ObjC only (requires workaround)
10.3.0-10.3.8	No	No	C++ requires SDK	C, ObjC only
10.3.9	No	No	C++ requires SDK	Yes
10.4	No	No	Yes	Yes
10.4 (Universal)	No	No	No	Yes

Remove libstdc++.a

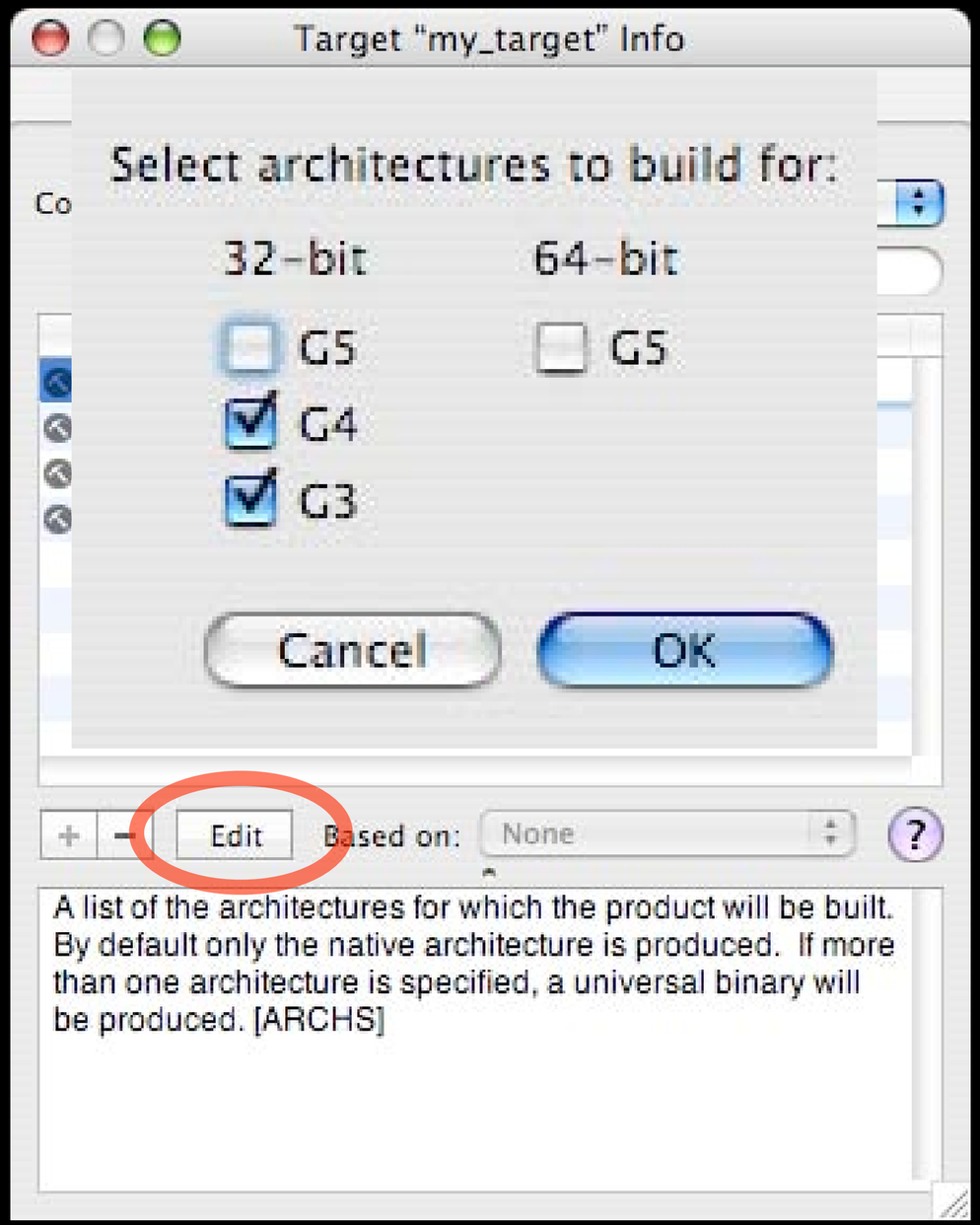


128-bit long double

- 53 extra bits of precision
- Part of the permanent API on Mac OS
- Available on 10.3.9 and later
- Compilers before 4.0 warn on every use of `long double`
- Not the same as `long double` on Intel

Better CPU Support

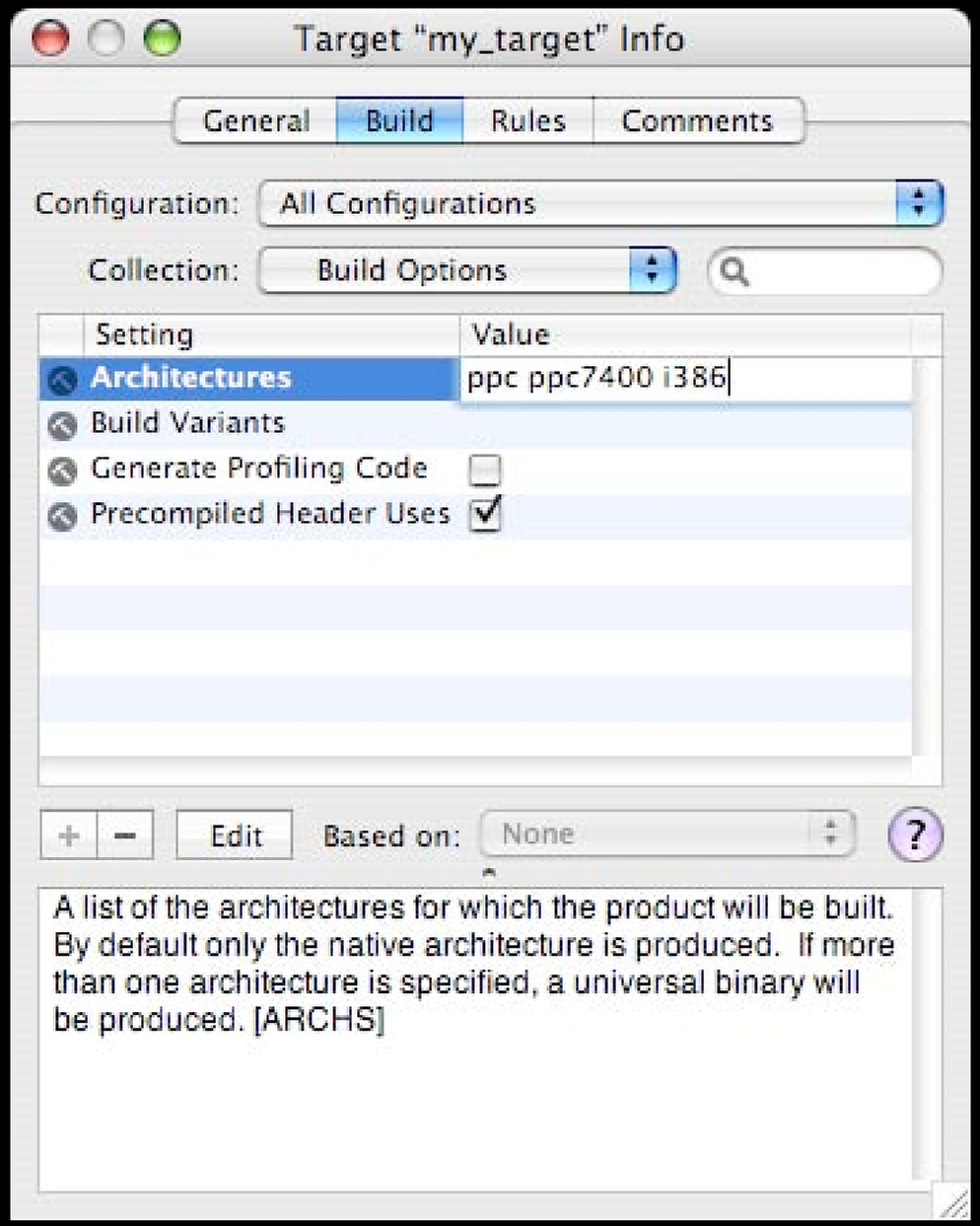
- Automatic vectorisation, starting to be useful for real applications
- Vectorised inline `memset` and `memcpy`



Better CPU Support

Xcode 2.0

- Universal executables for G3/G4/G5
- 64-bit



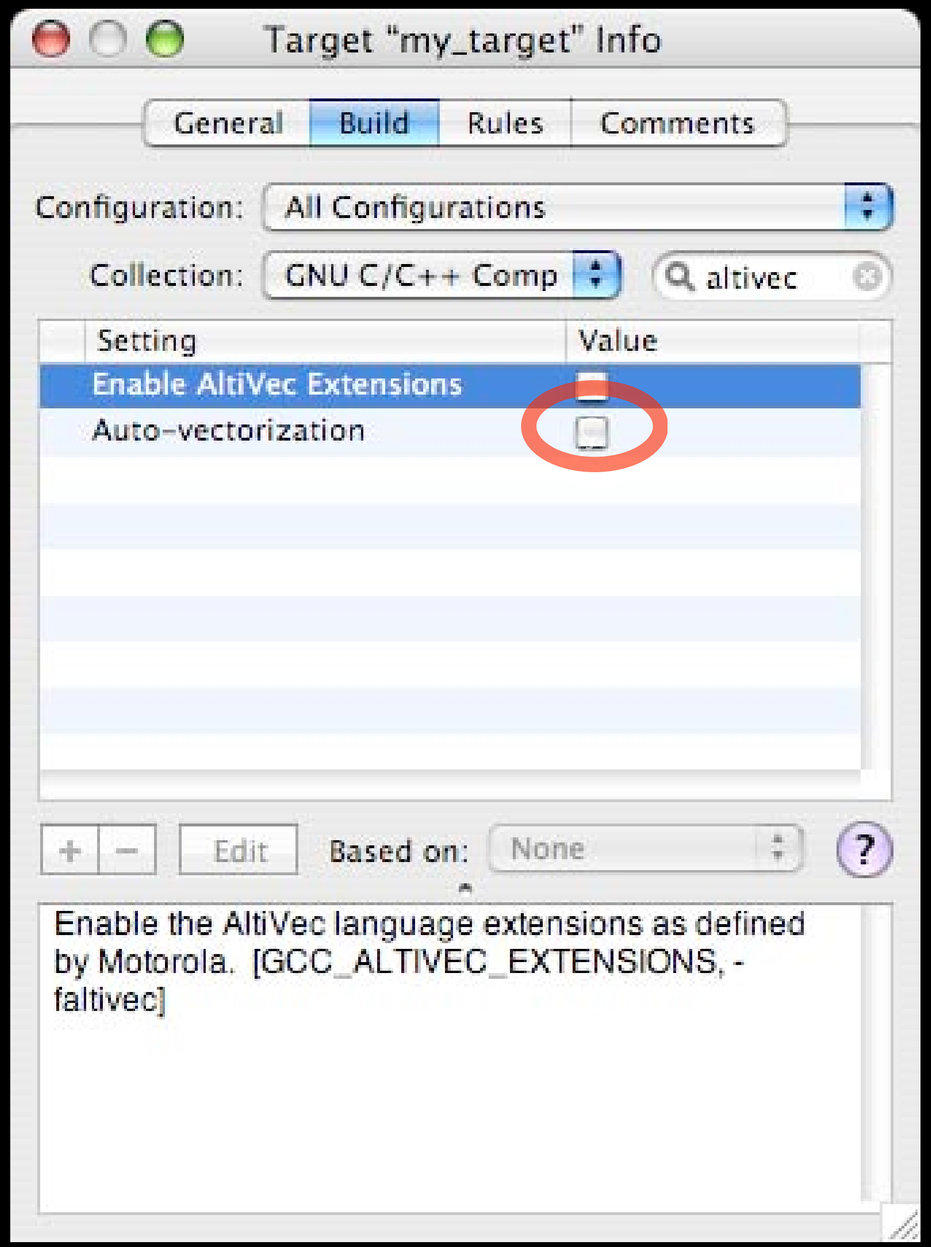
Better CPU Support

Xcode 2.1

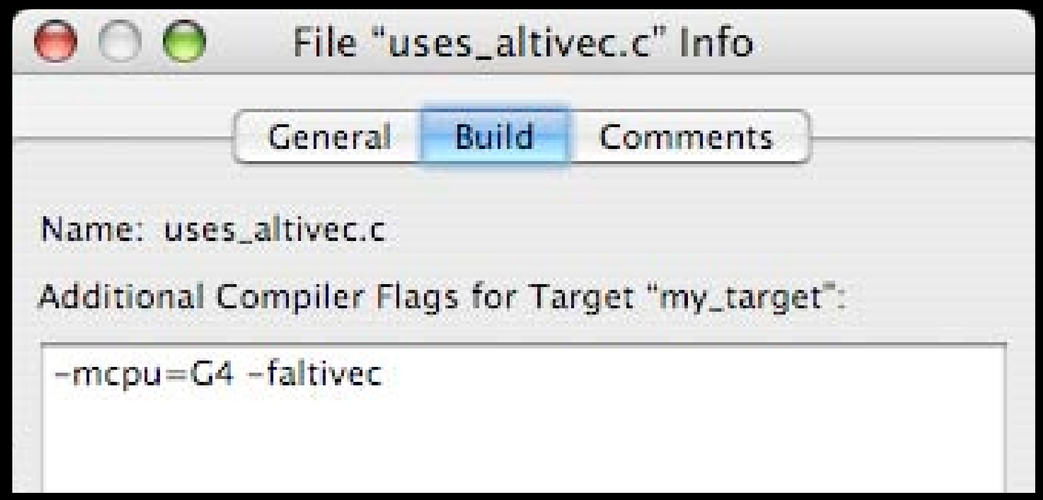
- In Xcode 2.1, the dialog only offers 32-bit G3 and Intel
- You may add other targets by hand, see `man 3 arch`

Checkbox	Architecture Name
PowerPC/G3 (32-bit)	ppc
G4 (32-bit)	ppc7400
G5 (32-bit)	ppc970
G5 (64-bit)	ppc64
Intel	i386

Better CPU Support



- Do use `-mcpu=G4` for G4-only object files



- Try not to use `-faltivec` for files that should run on G3

New and Better Warnings

- Read them! Fix them!
- Consider removing `-Werror` temporarily
- Enable `-Wstrict-selector-match` for Objective-C
- Especially watch for “if this code is reached, the program will abort” and verify that the code is never executed

Flags Removed

- `-Wno-precomp` (about missing old-style precomps)
- `-fcoalesce`, `-fweak-coalesced`, `-fcoalesce-templates`
- `--param max-inline-insns`, `--param min-inline-insns`

Better CodeWarrior Assembler Support

Xcode 2.0 and 2.1

- Many bugs fixed
- GCC now knows which registers are used

Objective-C++ Ivars Run Constructors

- You may use non-POD C++ types as members of Objective-C objects!

```
@interface intmap : NSObject {
        map<int, int> the_map;
}
```

- Requires new flag

Call C++ Default Ctors/Dtors in Objective-C ☑

- Requires runtime support available with Tiger

Symbol Visibility Improvements

- Move to the standard GCC syntax

```
private_extern int x;
```

```
__attribute__((visibility ("hidden")))
int x;
```

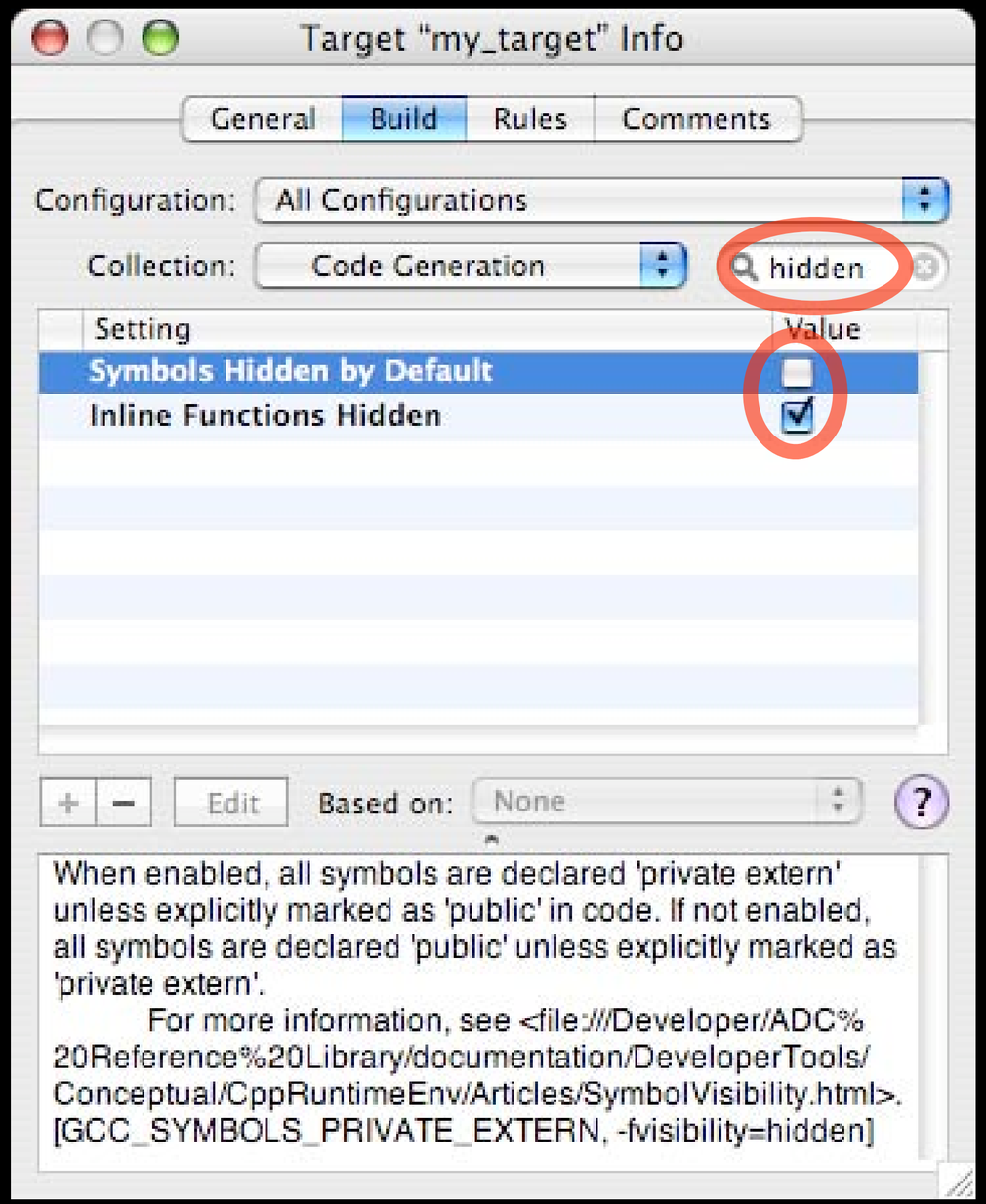
- Better yet, use hidden by default in frameworks

```
int x;
```

```
__attribute__((visibility ("default")))
int x;
```

- Now works on C++ classes!

Applications With Plugins/dynamic_cast



Many Bugs Fixed!

http://developer.apple.com/wwdc2005/

Dependent Name Lookup

```
template <typename T> class some_class {
  int class_local;
};
template <typename T> class another_class
    : public some_class <T> {
  int get_the_local () {
    return this->class_local;
  }
};
```

error: 'class_local' was not declared in this scope

Use of typename

```
template <class U> struct class1;

template <class T> struct class2 {
  class1<T>::type1> x;
};
```

error: to refer to a type member of a template parameter, use 'typename T::type1'

Use of template

```
template <class T1> struct A {
  template <int i> static void func1 ();
};

template <class T2> void func2() {
  A<T2>::func1<0>();
};
```

error: expected primary-expression before ')' token

In-Class Initialisation of static Members

```
class some_class {
  static const char * string = "string";
};
```

error: invalid in-class initialization of static data member of non-integral type 'const char*'

In-Class Initialisation of static Integer Members

```
class some_class {
  static const int one = 1;
};
```

Undefined symbols: some_class::one

Access Checking and Copy Constructors

```
class A {
  public: A();
  private: A (const A &);
};
void func1(const A &);
void func2() {
  A a1;
  func1 (A() a1);
}
```

error: 'A::A(const A&)' is private

Abstract Virtual Methods and NULL

```
class class1 {
  virtual int func1() = NULL;
};
```

error: expected ‘0’ before "__null"

Parentheses Around New Classname

```
class class1 { };
void func1() {
  class1 * c1p = new (class1)[100];
}
```

error: array bound forbidden after parenthesized type-id

Default Parameters

```
extern int func(int x = 3);

int func (int x = 3) {
  return x;
}
```

error: default argument given for parameter 1 of 'int func(int)'

No-Operation reinterpret_cast

```
int func1 (int x)
{
  return reinterpret_cast static_cast<int>(x);
}
```

error: invalid cast from type 'int' to type 'int'

offsetof Usage

```
struct class1 {
  int field1[100];
};

size_t get_field_offset (int index) {
  return offsetof (class1, field1[index]);
}
```

error: 'index' cannot appear in a constant-expression

Declarations in Namespaces

Xcode 2.1

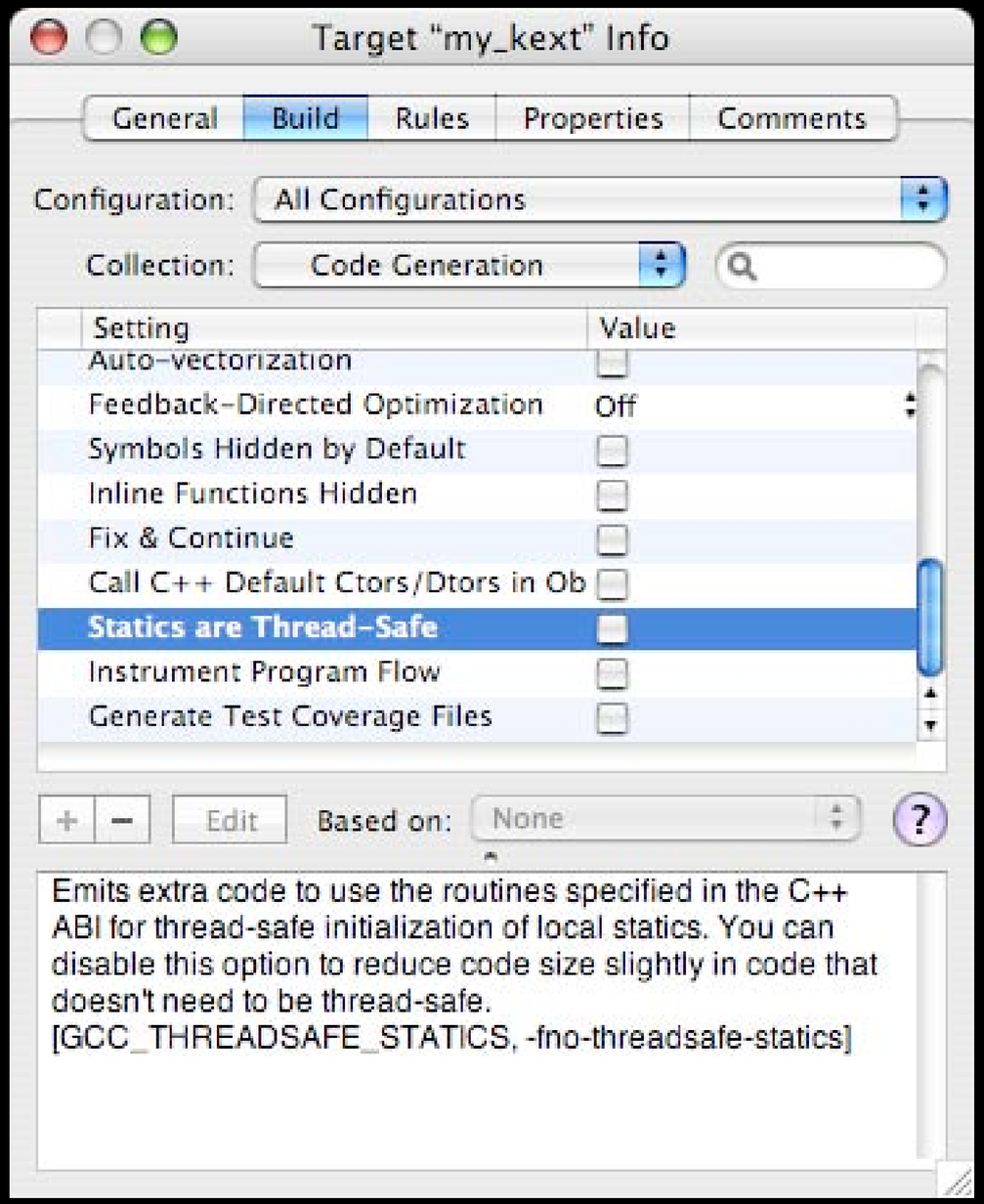
```
namespace mynames {
  void mynames::func1 ();
};
```

error: explicit qualification in declaration of ‘void mynames::func1()’

Kexts and static Local Variables

```
struct class1 {
  class1() { i = 1; };
  int i;
};
static class1 x;
int func() {
  static class1 x;
  return x.i;
}
```

Undefined symbols: ___cxa_guard_acquire ___cxa_guard_release



Kexts and static Local Variables

- Alternatively, use `-fno-threadsafe-statics`
- …but only if you're really sure it's safe!

Kexts and Member Function Pointers

```
interruptSource =
  IOInterruptEventSource::interruptEventSource(
    (OSObject*)this,
    OSMemberFunctionCast(IOInterruptEventAction,
      this,
      &UniNEnet::interruptOccurred),
  (IOService*)provider,
  (int)0 );
```

error: converting from `void (UniNEnet::*)(IOInterruptEventSource*, int)' to `void (*)(OSObject*, IOInterruptEventSource*, int)' in a kext. Use OSMemberFunctionCast() instead

Local static Function Declarations

```
static void some_other_function (int);

void some_function (void)
{
  static void some_other_function (int);
  some_other_function (10);
}
```

error: invalid storage class for function 'some_other_function'

static and extern

```
extern int ii;
static int ii;
```

error: static declaration of 'ii' follows non-static declaration

Labels at End of Block

```
void func1(void) {
  label:
}
```

error: label at end of compound statement

Lvalues!

```
char var1, var2;

void func1(int x)
{
  *(x < 0 ? &var1 : &var2) = 0;
  *(unsigned char *)&var1 = 200;
  *(var1, &var2) = 3;
}
```

error: invalid lvalue in assignment

Sequence Points

```
{
  func (i, i++);
  i++;
}
```

warning: operation on 'i' may be undefined

volatile and Casts

```
void func(void) {
  volatile int x = 0;
  while (*(volatile int *)x == 0)
    ;
}
```

Accidentally Excluding 4.0

```
#if __GNUC__ >= 3 || \
(__GNUC__ == 3 && __GNUC_MINOR__ >= 1)
```

Framework Header Searching

- Only the first found copy of a framework is searched
- `-I` and `-F` are not interchangeable

Other Changes

- Objective-C detects selector matching better
- `return;` in function supposed to return a value
- `return something;` in function returning `void`
- Using return value of function that returns `void`

FSF and Open Source

- The GCC 4.0 in Xcode 2.1 is based on the final FSF GCC 4.0
- Source code is available at
  http://darwinsource.opendarwin.org/WWDC2005/

Summary

- Use GCC 4.0! It's our best compiler yet
- You may have to adjust your code a little
- Most of these problems are rare and/or easily fixed

More Information

http://developer.apple.com/wwdc2005/

Documentation, Sample Code, and Other Resources

http://developer.apple.com/wwdc2005/

Related Sessions

Xcode Support	Lab Thursday 9:00-12:00
Development Tools Feedback Forum	Session 410 Friday 5:00PM

Who to Contact

Matthew Formica
Developer Tools and Cocoa Evangelist
mformica@apple.com

Q&A

Matthew Formica
Developer Tools and Cocoa Evangelist
mformica@apple.com

http://developer.apple.com/wwdc2005/

WWDC2005

Worldwide Developers Conference